

PENCARIAN JALUR TERPENDEK DALAM PROSES AUTODOCKING PADA MOBILE ROBOT MENGGUNAKAN ARTIFICIAL BEE COLONY

Safriudin Rifandi¹, Indra Adji Sulistijono¹, Son Kuswadi¹

¹ Politeknik Elektronika Negeri Surabaya, Kampus ITS,
Jl. Raya ITS, Keputih, Sukolilo, Kota Surabaya, Jawa Timur Indonesia

E-mail: safribuffon@pasca.student.pens.ac.id¹, indra@pens.ac.id², sonk@pens.ac.id³

Abstrak

Proses *autodocking* pada *mobile robot* merupakan sebuah kemampuan tambahan yang dapat dimiliki oleh sebuah robot. Untuk dapat melakukan proses tersebut, robot terlebih dahulu harus dapat mencari tempat *docking station* terdekat. Selain itu, robot membutuhkan pemandu agar dapat menuju ke *docking station* terdekat dengan tepat dan aman. Untuk mendapatkan jalur terdekat, dibutuhkan posisi terakhir robot saat mendeteksi baterai dalam keadaan akan habis. Posisi robot tersebut sebagaimana masukan bagi *artificial bee colony* untuk menghasilkan jalur terdekat yang dapat memandu robot menuju *docking station* terdekat. *Artificial bee colony* digunakan sebagai optimasi pencarian tempat *docking station* yang jarak tempuhnya terdekat atau terpendek dari posisi robot serta aman dari halangan atau *obstacle*. Dari hasil percobaan, metode *artificial bee colony* mampu untuk mendapatkan jalur terpendek menuju lokasi *docking station* serta jalur yang didapatkan juga aman dari halangan. Jumlah populasi lebah yang digunakan sangat mempengaruhi jarak terdekat yang didapatkan oleh metode *artificial bee colony*. Semakin banyak populasi yang digunakan, maka semakin dekat jarak yang didapatkan oleh metode *artificial bee colony*. Jumlah iterasi juga mempengaruhi jarak terdekat yang didapatkan oleh *artificial bee colony*. Semakin banyak jumlah iterasi yang digunakan, maka semakin dekat jarak yang didapatkan meskipun jumlah populasi yang digunakan sama.

Kata kunci: *autodocking*, *artificial bee colony*, *mobile robot*, *docking station*.

Abstract

Autodocking process at mobile robot is an additional capability of robot. Therefore the initial phase for applying this process, the robot should be able to find the nearest place of docking station. Then, the robot also require some guide for going to the docking station point accurately and safely. The shortest path to the docking station position can be obtained from last position of robot when its battery condition will run out. The last position of robot will be used as initial input for the artificial bee colony algorithm which can produce the shortest path of docking station position requirement. Artificial bee colony is used for optimizing the location search of docking station that having nearest distance to the robot position without any obstacle. From the results of the experiment, the artificial bee colony method was able to get the shortest path to the docking station location and the route obtained was also safe from obstacles. The number of bee populations used greatly influenced the closest distance obtained by the artificial bee colony method. The more population used, the closer the distance will be to the artificial bee colony method. The number of iterations also affects the closest distance obtained by artificial bee colony. The more number of iterations used, the closer the distance is obtained even though the population used is the same.

Keywords: *autodocking*, *artificial bee colony*, *mobile robot*, *docking station*

1. PENDAHULUAN

Keunggulan dalam teknologi robotika tak dapat dipungkiri telah lama dijadikan ikon kebanggaan negara-negara maju di dunia [1]. Banyak sekali robot yang telah dibuat dan beberapa dari robot tersebut sudah diproduksi secara massal untuk membantu kebutuhan manusia. Jenis robot yang dibuatpun bermacam-macam antara lain *non-mobile robot*, *mobile robot*, robot terbang, robot bawah air, dan robot *humanoid* [2].

Robot-robot tersebut membutuhkan sumber tegangan agar dapat bekerja. Sumber tegangan tersebut dapat berupa sumber tegangan AC atau sumber tegangan DC. Sumber tegangan AC biasa digunakan untuk jenis robot yang tidak dapat berpindah dari satu tempat ke tempat yang lain. Sehingga selama pasokan listrik masih ada, robot tersebut dapat terus bekerja tanpa ada batas waktu. Sedangkan sumber tegangan DC biasa digunakan untuk jenis robot yang dapat berpindah dari satu tempat ke tempat yang lain (*mobile robot*).

Pada jenis *mobile robot*, selama bekerja energi robot akan berkurang dan harus diisi sebelum baterai benar-benar habis. Maka pada level baterai tertentu robot harus segera melakukan pengisian baterai. Agar dapat kembali bekerja. Agar lebih efektif, pengisian baterai tersebut harus dapat dilakukan secara mandiri oleh robot atau secara otomatis.

Autodocking sendiri terdapat beberapa metode yang sering di gunakan untuk mengisi baterai. Metode-metode tersebut antara lain metode *docking station* [3], metode catenary, metode rel ketiga [4], metode wireless, dan metode penggantian baterai [5]. Dari metode-metode tersebut memiliki beberapa keunggulan dan kelemahan antara lain pada metode *docking station* memiliki keunggulan biayanya murah, bisa ditempatkan dimana saja dan membutuhkan area yang kecil. Sedangkan kelemahannya antara lain robot harus berhenti bekerja pada saat mengisi baterai [3]. Kemudian untuk metode catenary memiliki keunggulan yaitu robot dapat tetap berjalan hanya di wilayah tertentu yang terdapat saluran kabel listrik di atas robot. Untuk kelemahan metode catenary antara lain kabel listrik yang digunakan untuk mengisi baterai rawan terkena manusia yang dapat menyebabkan sengatan listrik. Sedangkan pada metode rel ketiga keunggulan dan kelemahan sama dengan metode catenary. Perbedaan hanya terletak pada material yang digunakan untuk mengalirkan listrik saat mengisi baterai yaitu berupa sebuah rel yang berada di samping robot [4].

Kemudian untuk metode wireless memiliki keunggulan antara lain tidak menggunakan kabel dalam proses pengisian baterai. Sedangkan kelemahannya antara lain membutuhkan biaya yang cukup besar untuk mengaplikasikannya dan sistem pengisian dengan wireless ini cukup rumit. Untuk metode penggantian baterai memiliki keunggulan antara lain membutuhkan waktu yang singkat untuk melakukan proses ini karena baterai langsung diganti dengan yang masih penuh. Sedangkan kelemahannya antara lain baterai tidak dapat diisi kembali [5]. Dari keunggulan

dan kelemahan yang dimiliki oleh metode-metode tersebut maka dipilihlah metode *docking station* untuk melakukan proses *autodocking* pada robot karena biayanya murah, bisa ditempatkan dimana saja dan membutuhkan area yang kecil. Selain itu, sistem *docking station* cukup mudah diimplementasikan.

Untuk dapat melakukan proses *autodocking*, robot terlebih dahulu harus dapat mencari tempat *docking station* terdekat. Selain itu, robot membutuhkan pemandu agar dapat menuju ke tempat *docking station* terdekat dengan tepat dan aman. Penelitian yang dilakukan oleh Alfin Junaedy, dkk (2017) [14] dapat dijadikan salah satu referensi. Pada penelitian tersebut, digunakan metode particle swarm optimization (PSO) pada proses *image processing* untuk mendeteksi sebuah benda.

Penelitian tentang *autodocking* ini telah dikembangkan oleh beberapa peneliti. Antara lain penelitian yang dilakukan K. H. Kim, dkk (2005) [6] menggunakan *scanning* area pemancar infra-merah untuk menentukan posisi *docking* dan membaca kode infra-merah yang dipancarkan *docking* untuk menentukan posisi tengah dari *docking station*. Kemudian penelitian oleh Se-Gon Roh, dkk (2008) [7] juga menggunakan Sonar dan infra-merah untuk menemukan tempat *docking station* serta menggunakan kompensasi kesalahan untuk menentukan posisi tengah *docking station*. Demikian juga penelitian yang dilakukan Rudy Purwanto dan Djoko Purwanto (2009) [8] juga menggunakan metode *scanning* dengan *photodiode* dan *infrared* untuk menemukan lokasi *dockingstation* dan penelitian oleh K. Varun Raj, dkk (2007) [9] menggunakan dua buah *infrared* sebagai petunjuk bagi robot untuk menuju tempat *docking*. Sedangkan penelitian yang dilakukan N. Sravanthi dan J. Krishna Chaithanya (2014) [10] menggunakan metode *path planning* dengan sensor infra merah untuk menuju tempat *docking station*. Kuo-Lan Su, dkk (2014) [11] juga melakukan penelitian dengan menggunakan komunikasi antara robot dengan *docking station* untuk

mengetahui posisi *docking station* dan menggunakan *laser range finder* untuk menemukan petunjuk yang memandu robot menuju tempat *docking*.

Dari penelitian-penelitian sebelumnya terdapat beberapa kekurangan yaitu masih menggunakan sensor infra-merah untuk menemukan lokasi *docking* serta hanya menggunakan satu buah *docking station*. Dari kekurangan tersebut maka perlu dibuat sebuah cara atau sistem untuk *autodocking* yang lebih efektif dan efisien dengan menggunakan metode yang lebih baik serta menggunakan lebih dari satu *docking station*.

2. METODE

A. Artificial Bee Colony

Artificial bee colony algorithm merupakan salah satu dari beberapa algoritma *Swarm Intelligence* atau juga disebut *Swarm-based Optimisation (SoA)*. Kebanyakan algoritma *Swarm-based* meniru pola tingkah laku koloni hewan untuk pencarian makanan.

Meskipun sama-sama berfungsi untuk pencarian solusi, algoritma *SoA* berbeda dengan algoritma *Direct Search* pada umumnya. Algoritma pada kategori ini menggunakan kumpulan populasi untuk membangkitkan solusi-solusi ketimbang hanya mencari satu solusi saja seperti yang dilakukan algoritma *Direct Search*. Populasi solusi ini kemudian diproses dalam iterasi dan pada akhirnya akan menghasilkan populasi solusi terbaik [12].

Dalam *artificial bee colony*, koloni lebah buatan mengandung tiga kelompok lebah yaitu *employed bees* terkait dengan sumber makanan tertentu, *onlooker bees* menyaksikan tarian lebah yang bekerja dalam sarang untuk memilih sumber makanan, dan *scout bees* mencari sumber makanan secara acak. Keduanya *onlooker bees* dan *scout bees* juga disebut *unemployed bees*. Awalnya, semua posisi sumber makanan ditemukan oleh *scout bees*. Setelah itu, nektar dari sumber makanan dieksploitasi oleh *employed bees* dan *onlooker bees*, eksploitasi yang terus-menerus ini pada

akhirnya akan menyebabkan mereka menjadi kelelahan.

Kemudian, *employed bees* yang mengeksploitasi sumber makanan yang kelelahan menjadi *scout bees* mencari sumber makanan lanjut sekali lagi. Dengan kata lain, *employed bees* yang sumber makanan telah habis menjadi *scout bees*. Dalam *artificial bee colony*, posisi sumber makanan merupakan solusi untuk masalah dan jumlah nektar dari sumber makanan sesuai dengan kualitas (fitness) dari solusi terkait. Jumlah lebah dipekerjakan sama dengan jumlah sumber makanan (solusi) karena setiap lebah yang digunakan dikaitkan dengan satu dan hanya satu sumber makanan. Sedangkan untuk skema umum algoritma *Artificial Bee Colony* dapat dilihat pada skema berikut ini [13]:

- Fase inisialisasi
- Ulangi
 - Fase *employed bees*
 - Fase *onlooker bees*
 - Fase *scout bees*
 - Ingat solusi terbaik yang telah dicapai
- Ulangi sampai kriteria penghentian tercapai

A.1. Fase Inisialisasi

Semua vektor dari populasi sumber makanan diinisialisasi oleh *scout bees* dan parameter kontrol ditetapkan. Karena setiap sumber makanan adalah vektor solusi untuk masalah optimasi. Persamaan (1) dapat digunakan untuk fase inisialisasi [13].

$$X_{mi} = l_i + rand(0,1) * (u_i - l_i) \quad (1)$$

Dimana :

- l_i = Batas bawah dari parameter
- u_i = Batas atas dari parameter
- X_{mi} = Populasi sumber makanan

A.2. Fase Employed Bees

Employed bees mencari sumber makanan baru yang memiliki jumlah nektar lebih

banyak di daerah sumber makanan yang diingat oleh *Employed bees*. Setelah mereka menemukan sumber makanan di dekat sumber makanan semula, selanjutnya mereka mengevaluasi nilai *fitness* dari sumber makanan yang baru. Misalnya mereka dapat menentukan sumber makanan di dekat sumber makanan semula dengan menggunakan Persamaan (2) [13].

$$V_{mi} = X_{mi} + r_i(X_{mi} - X_{ki}) \quad (2)$$

Dimana :

X_k = Sumber makanan yang dipilih secara acak

i = Indeks parameter yang dipilih secara acak

m = Nomor acak dalam rentang $(-a, a)$

V_{mi} = sumber makanan baru

Setelah memproduksi sumber makanan yang baru, nilai *fitness* dihitung dan dipilih antara V_m dan X_m . Nilai *fitness* dari solusi dapat dihitung dengan Persamaan (3) [13].

$$fit_m(\vec{X}_m) = \begin{cases} \frac{1}{1 + f_m(\vec{X}_m)} & \text{jika } f_m(\vec{X}_m) \geq 0 \\ \frac{1}{1 + abs(f_m(\vec{X}_m))} & \text{jika } f_m(\vec{X}_m) < 0 \end{cases} \quad (3)$$

Dimana :

$fit_m(\vec{X}_m)$ = Nilai fungsi tujuan dari solusi

A.3. Fase Onloker Bees

Unemployed bees terdiri dari dua kelompok lebah : *onloker bees* dan *scout bees*. *Employed bees* berbagi informasi sumber makanan mereka dengan *onloker bees* menunggu di dalam sarang dan kemudian *onloker bees* akan memilih sumber makanan mereka tergantung pada informasi ini. Dalam ABC, *onloker bees* memilih sumber makanan tergantung pada nilai-nilai probabilitas dihitung dengan menggunakan nilai-nilai kebugaran yang

disediakan oleh *Employed bees*. Untuk tujuan ini, teknik seleksi berdasarkan kebugaran dapat digunakan, seperti metode *roulette wheel*. Nilai probabilitas yang dipilih oleh *onloker bees* dapat dihitung dengan Persamaan (4) [13].

$$p_m = \frac{fit_m(\vec{X}_m)}{\sum_{m=1}^{SN} fit_m(\vec{X}_m)} \quad (4)$$

Dimana :

p_m = Nilai probabilitas

$fit_m(\vec{X}_m)$ = Nilai fungsi tujuan dari solusi

Setelah sumber makanan untuk *onloker bees* telah dipilih, sumber makanan di dekatnya dapat dihitung dengan persamaan (2) dan nilai *fitness* dapat dihitung. Seperti pada fase *employed bees*, pilihan ditetapkan antara V_m dan X_m . Oleh karena itu, lebih banyak *employed bees* yang direkrut untuk sumber makanan yang memiliki nektar lebih banyak [13].

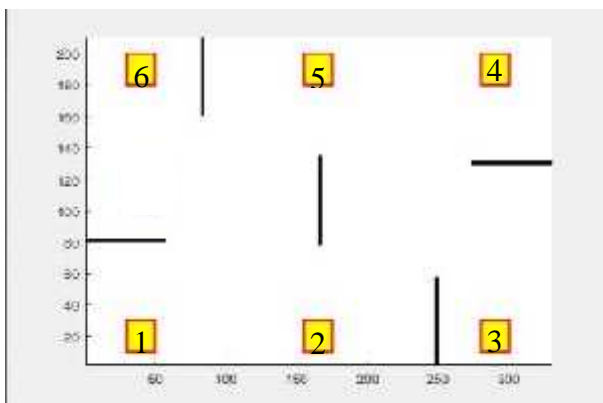
A.4. Fase Scout Bees

Unemployed bees yang memilih sumber makanan mereka secara acak disebut *scout bees*. *Employed bees* yang solusinya tidak dapat ditingkatkan melalui jumlah yang telah ditetapkan, ditentukan oleh pengguna dari algoritma ABC dan disebut "batas" atau "kriteria penghentian" di sini, *employed bees* menjadi *scout bees* dan solusi yang lama mereka tinggalkan. Kemudian, para pengintai dikonversi mulai mencari solusi baru secara acak. Misalnya, jika solusi telah ditinggalkan, solusi yang baru ditemukan oleh *scout bees* yang berasal dari *employed bees* yang bisa dihitung dengan Persamaan (1) [13].

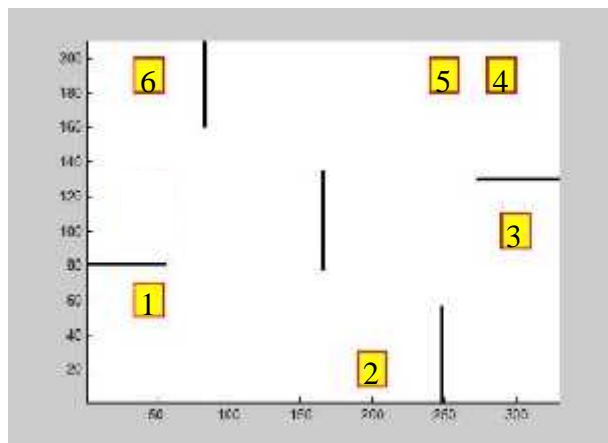
3. HASIL DAN PEMBAHASAN

Pada bagian ini, dibahas tentang hasil dari percobaan pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Percobaan dilakukan dalam beberapa skenario

atau model pengujian antara lain dengan merubah jumlah populasi yang digunakan, merubah lokasi dari *docking station* dan merubah lokasi robot. Area percobaan menggunakan 2 bentuk area dengan posisi *docking station* dan posisi robot yang berbeda. Untuk luas area percobaan memiliki ukuran panjang 330 cm dan lebar 210 cm. Gambar bentuk area percobaan dapat dilihat pada Gambar 1 dan Gambar 2. Pada Gambar 1, lokasi *docking station* berada pada koordinat (40, 20), (165, 20), (290, 20), (40, 190), (165, 190), dan (290, 190). Sedangkan pada Gambar 2 lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Dalam percobaan ini digunakan iterasi sebanyak maksimal 100 kali untuk setiap jalur dan menggunakan variasi jumlah populasi sebanyak maksimal 100 populasi.



Gambar 1. Area percobaan ke-1 pencarian jalur terpendek dengan metode *artificial bee colony*.



Gambar 2. Area percobaan ke-2 pencarian jalur terpendek dengan metode *artificial bee colony*.

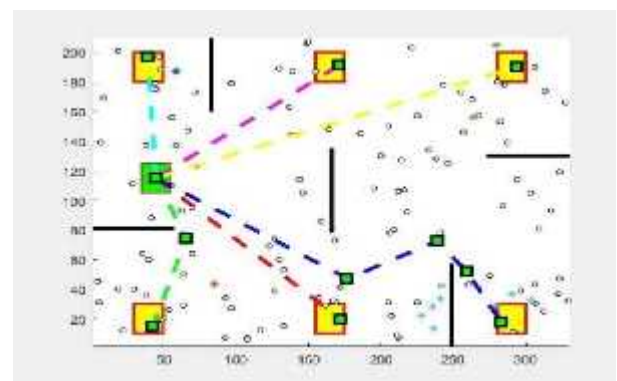
Keterangan gambar :

- : *Docking station*
- : Robot
- : Jalur dari robot ke *docking station* (setiap jalur memiliki warna yang berbeda)
- : Halangan (*obstacle*)

A. Percobaan Dengan Jumlah Populasi 20 Pada Area Percobaan ke-1

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 20. Sedangkan lokasi *docking station* berada pada koordinat (40, 20), (165, 20), (290, 20), (40, 190), (165, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 3. Posisi robot berada pada koordinat (45, 115).

Dari Gambar 3 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna biru muda. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 81,80 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,11 menit.



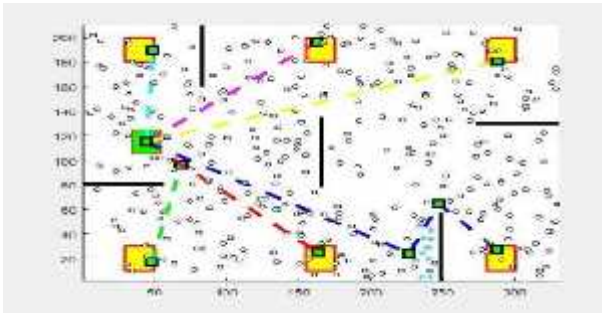
Gambar 3. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 20 pada area percobaan ke-1.

B. Percobaan Dengan Jumlah Populasi 60 Pada Area Percobaan ke-1

Pada percobaan ini menguji proses pencarian jalur terpendek dengan

menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 60. Sedangkan lokasi *docking station* berada pada koordinat (40, 20), (165, 20), (290, 20), (40, 190), (165, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 4. Posisi robot berada pada koordinat (45, 115).

Dari Gambar 4 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna biru muda. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 74,43 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,38 menit.



Gambar 4. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 60 pada area percobaan ke-1.

C. Percobaan Dengan Jumlah Populasi 100 Pada Area Percobaan ke-1

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 100. Sedangkan lokasi *docking station* berada pada koordinat (40, 20), (165, 20), (290, 20), (40, 190), (165, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 5. Posisi robot berada pada koordinat (45, 115).

Dari Gambar 5 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna biru muda pada Gambar 20. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju

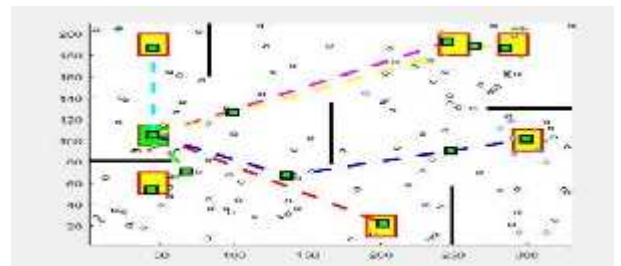
docking station nomor 6 adalah 68,85 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 2,16 menit.



Gambar 5. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 100 pada area percobaan ke-1.

D. Percobaan Dengan Jumlah Populasi 20 Pada Area Percobaan ke-2

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 20. Sedangkan lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 6. Posisi robot berada pada koordinat (45, 115).

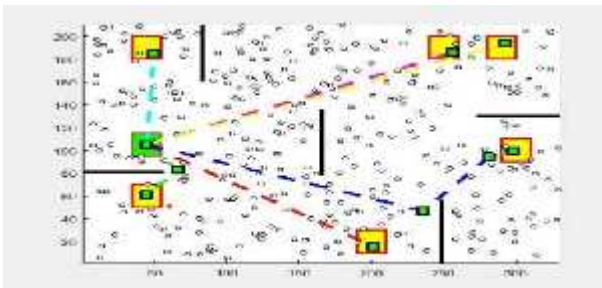


Gambar 6. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 20 pada area percobaan ke-2.

Dari Gambar 6 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna hijau. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 1. Jarak yang didapat pada jalur menuju *docking station* nomor 1 adalah 70,61 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,28 menit.

E. Percobaan Dengan Jumlah Populasi 60 Pada Area Percobaan ke-2

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 60. Sedangkan lokasi *docking station* berada pada koordinat (45 , 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 7. Posisi robot berada pada koordinat (45, 115).



Gambar 7. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 60 pada area percobaan ke-2.

Dari Gambar 7 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna hijau. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 1. Jarak yang didapat pada jalur menuju *docking station* nomor 1 adalah 61,58 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,10 menit.

F. Percobaan Dengan Jumlah Populasi 100 Pada Area Percobaan ke-2

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 100. Sedangkan lokasi *docking station* berada pada koordinat (45 , 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 8. Posisi robot berada pada koordinat (45, 115).

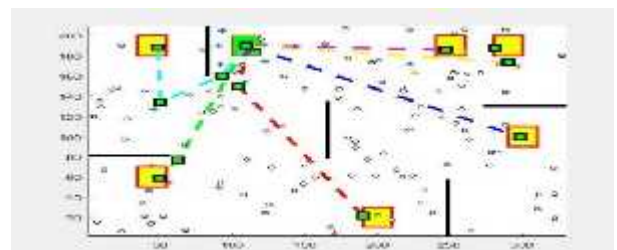
Dari Gambar 8 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna hijau. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 1. Jarak yang didapat pada jalur menuju *docking station* nomor 1 adalah 59,04 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 0,59 menit.



Gambar 8. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 100 pada area percobaan ke-2.

G. Percobaan Dengan Jumlah Populasi 20 Pada Area Percobaan ke-2 dengan posisi robot berpindah

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 20. Sedangkan lokasi *docking station* berada pada koordinat (45 , 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 9. Posisi robot berada pada koordinat (110, 190).



Gambar 9. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 20 pada area percobaan ke-2 dan posisi robot berpindah.

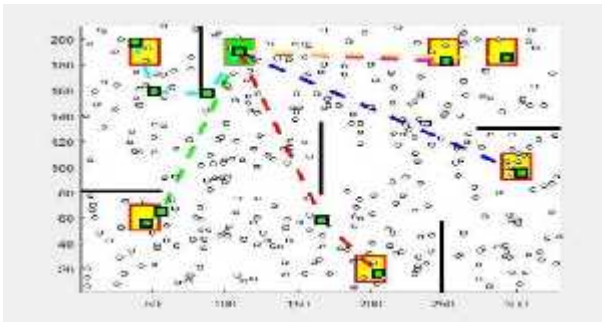
Dari Gambar 9 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna

ungu. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 5. Jarak yang didapat pada jalur menuju *docking station* nomor 5 adalah 137,58 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,27 menit.

H. Percobaan Dengan Jumlah Populasi 60 Pada Area Percobaan ke-2 dengan posisi robot berpindah

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 60. Sedangkan lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 10. Posisi robot berada pada koordinat (110, 190).

Dari Gambar 10 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna biru muda. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 116,56 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,02 menit.



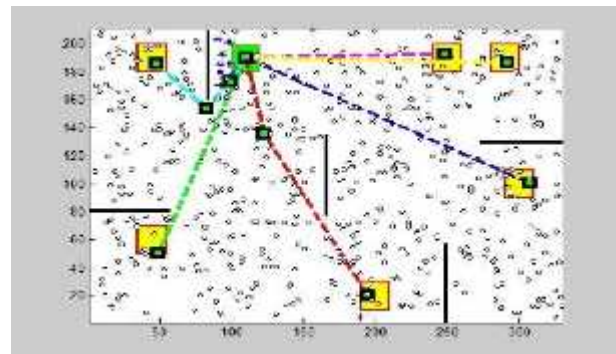
Gambar 10. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 60 pada area percobaan ke-2 dan posisi robot berpindah.

I. Percobaan Dengan Jumlah Populasi 100 Pada Area Percobaan ke-2 dengan posisi robot berpindah

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*.

Jumlah populasi lebah buatan yang digunakan berjumlah 100. Sedangkan lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 11. Posisi robot berada pada koordinat (110, 190).

Dari Gambar 11 dapat dilihat bahwa jalur yang terpendek adalah jalur dengan warna biru muda. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 93,57 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 2,18 menit.



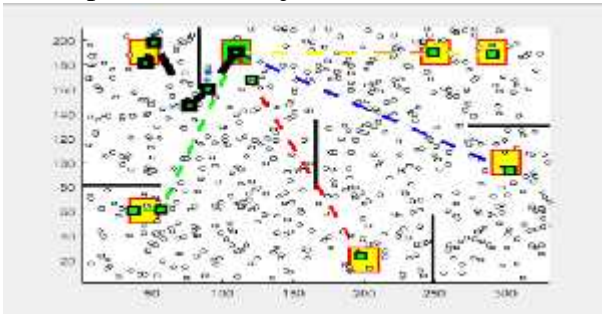
Gambar 11. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 100 pada area percobaan ke-2 dan posisi robot berpindah.

J. Percobaan Dengan Jumlah Populasi 100 Pada Area Percobaan ke-2 Dengan Jumlah Iterasi 20 kali

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 100. Sedangkan jumlah iterasi yang digunakan 20 kali. Untuk lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 12. Posisi robot berada pada koordinat (110, 190).

Dari Gambar 12 dapat dilihat bahwa dengan menggunakan jumlah iterasi 20 kali, jalur yang terpendek adalah jalur dengan warna hitam.

Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 131,12 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 1,10 menit.



Gambar 12. Hasil jalur yang didapat metode *artificial bee colony* dengan populasi 100 pada area percobaan ke-2 dengan jumlah iterasi 20 kali.

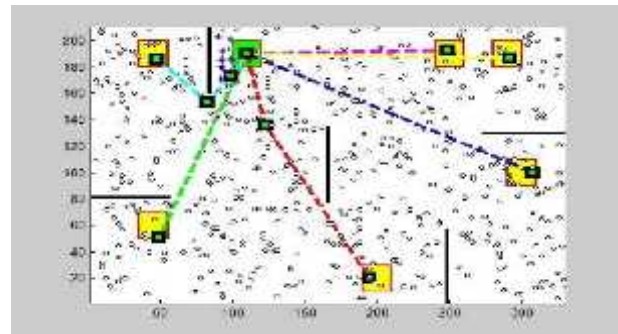
K. Percobaan Dengan Jumlah Populasi 100 Pada Area Percobaan ke-2 Dengan Jumlah Iterasi 40 kali

Pada percobaan ini menguji proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*. Jumlah populasi lebah buatan yang digunakan berjumlah 100. Sedangkan jumlah iterasi yang digunakan 40 kali. Untuk lokasi *docking station* berada pada koordinat (45, 60), (200, 20), (300, 100), (45, 190), (250, 190), dan (290, 190). Hasil simulasi proses pencarian jalur terpendek dengan menggunakan *artificial bee colony* dapat dilihat pada Gambar 13. Posisi robot berada pada koordinat (110, 190).

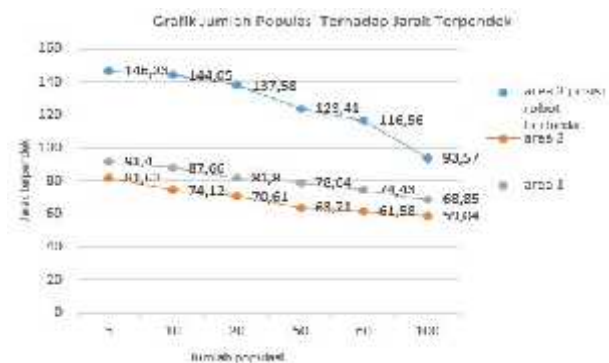
Dari Gambar 13 dapat dilihat bahwa dengan menggunakan jumlah iterasi 40 kali, jalur yang terpendek adalah jalur dengan warna biru muda. Sedangkan *docking station* yang terpilih adalah *docking station* nomor 6. Jarak yang didapat pada jalur menuju *docking station* nomor 6 adalah 93,57 cm. Sedangkan waktu komputasi yang dibutuhkan untuk mendapatkan seluruh jalur adalah 2,18 menit.

Setelah dilakukan beberapa percobaan pada proses pencarian jalur terpendek dengan menggunakan metode *artificial bee colony*, pada Gambar 14 dapat dilihat grafik perbedaan

populasi yang digunakan dengan jarak terdekat yang didapatkan oleh *artificial bee colony*.



Gambar 13. Hasil jalur yang didapat untuk *docking station* nomor 6 dengan populasi 100 pada area percobaan ke-2 dengan jumlah iterasi 40 kali.



Gambar 14. Grafik antara jumlah populasi dengan jarak terpendek (cm).



Gambar 15. Grafik antara jumlah iterasi dengan jarak terpendek (cm).

Dengan menggunakan populasi 100, jarak terdekat yang didapatkan lebih pendek dibandingkan jika menggunakan populasi 60, 50, 20, 10 atau 5. Sedangkan pada Gambar 15 dapat dilihat grafik perbedaan jumlah iterasi yang digunakan dengan jarak terdekat yang

didapatkan. Dengan jumlah iterasi 100 kali, jarak terdekat yang didapatkan lebih pendek dibandingkan dengan jika menggunakan jumlah iterasi 50, 40, 20, 10 atau 5 kali.

4. KESIMPULAN

Penelitian ini bertujuan untuk mencari jalur terpendek dalam proses *autodocking* pada *mobile robot* menggunakan *artificial bee colony*. Dari hasil percobaan, metode *artificial bee colony* mampu untuk mendapatkan jalur terpendek menuju lokasi *docking station* serta jalur yang didapatkan juga aman dari halangan. Jumlah populasi lebah yang digunakan sangat mempengaruhi jarak terdekat yang didapatkan oleh metode *artificial bee colony*. Semakin banyak populasi yang digunakan, maka semakin dekat jarak yang akan didapatkan oleh metode *artificial bee colony*. Jumlah iterasi juga mempengaruhi jarak terdekat yang didapatkan oleh *artificial bee colony*. Semakin banyak jumlah iterasi yang digunakan, maka semakin dekat jarak yang didapatkan meskipun jumlah populasi yang digunakan sama. Aplikasi penelitian ini nantinya dapat diterapkan pada *mobile robot* agar lebih efisien dalam melakukan proses pengisian baterai.

DAFTAR PUSTAKA

- [1] E. Pitowarno, *Robotika : Desain, Kontrol, dan Kecerdasan Buatan*, Yogyakarta, Indonesia : ANDI, 2006.
- [2] R. Supriyanto, Hustinawati, R. W. Nugraini, A. B. Kurniawan, Y. Permadi, and A. Sa'ad, *Robotika*, Jakarta, Indonesia : Universitas Gunadarma, 2010.
- [3] C. Ricaud and P. Vollet, "Connection method for charging systems a key element for electric vehicle", Schneider Electric Conference, France, 2010, pp. 1-11.
- [4] H. Maghfiroh. (view August. 2017). Dunia kereta-sistem propulsi KRL [Online]. Available : <http://www.keretalistrik.com/2016/07/dunia-kereta-sistem-propulsi-krl.html>.
- [5] CLP Power, *Typical Electric Vehicle Charging Methods*, Sham Shui Po, Hong Kong : CLP Power, 2013.
- [6] K. H. Kim, H. D. Choi, S. Yoon, K. W. Lee, H. S. Ryu, C. K. Woo, and Y. K. Kwak, "Development of docking system for mobile robots using cheap infrared sensors", 1st International Conference on Sensing Technology, New Zealand, 2005, pp. 287-291.
- [7] S. Roh, J. H. Park, Y. H. Lee, Y. K. Song, K. W. Yang, M. Choi, H. S. Kim, H. Lee, and H. R. Choi, "Flexible docking mechanism with error-compensation capability for auto recharging system of mobile robot", International Journal of Control, Automation, and Systems, Vol. 6, 2008, pp. 731-739.
- [8] R. Purwanto and D. Purwanto, *Rancang Bangun Sistem Autodocking Untuk Pengisian Baterai Pada Mobile Robot*, Surabaya, Indonesia : Institut Teknologi Sepuluh Nopember Surabaya, 2009.
- [9] K. Varun Raj, K. Patil, D. V. K. Kariappa, and A. M. Jakati, "A Beacon-based docking system for an autonomous mobile robot", National Conference on Mechanisms and Machines (NaCoMM07), India, 2007, pp. 1-7.
- [10] N. Sravanthi and J. Krishna Chaithanya, "A Surveillance robot with automatic charging unit for home security", International Journal of Science, Engineering and Technology Research, Vol. 3, 2014, pp. 2400-2404.
- [11] K. L. Su, Y. L. Liao, S. P. Lin, and S. F. Lin, "An Interactive auto-recharging system for mobile robots", International Journal of Automation and Smart Technology, Vol. 4, 2014, pp. 43-53.
- [12] V. Christmantara, *Studi Tentang Bees Algorithm Dan Penerapannya*, Surabaya, Indonesia : Institut Sains Terapan dan Teknologi Surabaya, 2011.
- [13] D. Karaboga. (view Augst. 2017). Artificial Bee Colony Algorithm. SCHOLARPEDIA[Online]. Available: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm#Karaboga07b.
- [14] A. Junaedy, I. A. Sulistijono, N. Hanafi, "Particle Swarm Optimization for Coconut Detection in a Coconut Tree Plucking Robot". International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Indonesia, 2017, pp. 182-187.